

Introduction to Computational Physics
PHYS 250 (Autumn 2018) – Lecture 2

David Miller

Department of Physics and the Enrico Fermi Institute
University of Chicago

October 4, 2018

Outline

1 *Introduction*

2 *Software*

Computational Physics (PHYS 250)

Course Description PHYS 250 ([link to Course Catalog](#))

This course introduces the use of computers in the physical sciences. After an introduction to programming basics, we cover numerical solutions to fundamental types of problems, including cellular automata, artificial neural networks, computer simulations of complex systems, and finite element analysis. Additional topics may include an introduction to graphical programming, with applications to data acquisition and device control.

There are an infinite number of paths that we might follow and still not deviate from this description. I therefore would like to lay out some of the principles that will guide me, and us, in how we navigate through those many possibilities.

Outline

1 *Introduction*

2 *Software*

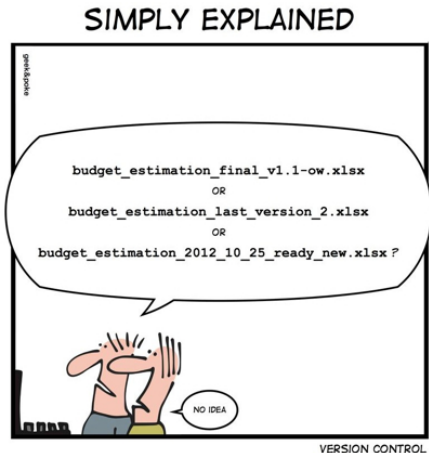
Version control

- The most important message of this slide is simple...**Use a software version control system for all of your code**
 - And that means now...not tomorrow or next week
 - Because if you wait until you need it, it will be too late

Version control

- The most important message of this slide is simple... **Use a software version control system for all of your code**

- And that means now...not tomorrow or next week
- Because if you wait until you need it, it will be too late



A brief history of version control

- **The first version control systems were designed to be used on large systems where everyone logged into the same machine**
 - They tracked code on the same filesystem where it lived (e.g., in a subdirectory)
 - SCCS and RCS are examples
- **Then client-server systems were developed, so that developers could work on their own machines**
 - Checking code into a central server to share and collaborate
 - CVS and SVN are examples
- **More recently distributed version control systems have arisen**
 - These are decentralised, so everyone has a complete copy of the repository
 - Gives a lot of freedom to developers to share and merge as they like, so liked very much by the open source community
 - git, mercurial and bit keeper are examples

git, GitHub, & GitLab

<https://git-scm.com>, <https://github.com>, <https://about.gitlab.com>

- **git is the most popular open source version control system**
 - can host huge projects (Linux Kernel, LHC experiment software, etc)
 - scales very well and it's extremely fast and powerful
 - very flexible (= complex)
- **Distributed version control systems (git) are great, but they're made even better by using a social coding site (GitHub or GitLab)**
- **These sites allow developers:**
 - browse code easily
 - compare different versions
 - take copies (a.k.a. fork)
 - offer patches back to upstream repositories
 - And discuss and review these patches before acceptance
 - even build websites
- **The best known social coding site is GitHub, but there are others, e.g. BitBucket and GitLab**
 - Familiarity with **git/GitHub/GitLab** will serve you well, trust me

GitHub & GitLab resources

GitHub is a free resource as long as your code remains public (you can pay for private repositories). The Physical Sciences Division (PSD) at UChicago hosts a **private GitLab** repository.

• <https://psdcomputing.uchicago.edu/page/psd-repo>



PSD Repo



PSD Repo is a software source code repository managed by the PSD Computing office

UC LDAP	Standard
UC LDAP Username	
<input type="text" value="johndoe"/>	
Password	
<input type="password" value="....."/>	
<input type="checkbox"/> Remember me	
<input type="button" value="Sign in"/>	

PHYS 250 *GitHub*

<https://github.com/UChicagoPhysics/PHYS250>

Course materials are hosted in the **GitHub** UChicagoPhysics repository

UChicagoPhysics / PHYS250

Watch 0 Star 0 Fork 0

Code Issues 0 Pull requests 0 Projects 0 Wiki Insights Settings

University of Chicago PHYS 250 Computational Physics software repository

Manage topics

15 commits 1 branch 0 releases 1 contributor

Branch: master New pull request

Create new file Upload files Find file Clone or download

fizisist Update example		Latest commit 0f09be5 25 minutes ago
Examples	Update example	25 minutes ago
LearningGoals	Update Learning Goals	22 hours ago
Slides	Update Lecture 1 Slides	an hour ago
Syllabus	Updates to syllabus	22 hours ago
.gitignore	Update slides for day 1	3 days ago
README.md	Update README.md	3 days ago

- Slides (e.g. *these!*), syllabi, learning goals, and code examples
- Stable versions will be cross-posted to **Canvas** as well.
- Homework submission will be done via **GitHub** (*instructions to come*)

Linux “shell”

- We will be using an interface to Linux called a “shell”
- It is a command-line interpreter : you type, it executes
- Two major options are `bash` (as in, smash) and `csh` (like “sea shell”, modern version is “tcsh”, “tea sea shell”)
 - Only real difference: environment variables syntax
 - `bash`: `export X=value`
 - `csh`: `setenv X value`

Shell basics

Listing directory contents : `ls`, like “list”

```
> ls
```

```
Examples/ LearningGoals/ README.md Slides/  
Syllabus/ global.sty
```

Copy: `cp`

```
> cp stuff.txt stuff1.txt
```

Where am I?: `pwd`, `cd`

```
> pwd
```

```
/ComputationalPhysics/PHYS250/PHYS250-Autumn2018
```

```
> ls
```

```
Examples/ LearningGoals/ README.md Slides/ Syllabus/
```

```
> cd Examples/
```

```
> ls
```

```
HelloGaussian.ipynb
```

```
HelloGaussian.py
```

```
Introduction_to_Jupyter_Notebooks_and_Python.ipynb
```

Hello world!

Interactive in the python interpreter

```
python
>>> print "hello world"
hello
>>> CTRL-d  # to exit python
```

From a script (containing the above print line):

```
python helloworld.py
```

Self-running script:

```
#!/usr/bin/env python
# This script prints hello to the screen
print "hello world"
```

```
chmod +x helloworld.py
./helloworld.py
hello world
```

Lists (I)

In my opinion, python's great advantage is **list comprehension**.

List basics

```
v = []      # empty list
v = list()  # empty list
v = [ 1, 2, 4, 5 ] ; v = [ 'a', 'b', 'c' ]
v = range(4,10,2) # results in [ 4,6,8 ]
v = [ 4, 2.5, 'Hi', [ 1,3,5 ] ] # can mix types
```

Append elements

```
>>> v.append( 70 )
>>> print v
```

Concatenation

```
>>> v += [ 'some', 'more', 'elements' ]
>>> v # shows the object
```

Removal of elements

```
>>> v.remove(2.5)
>>> del v[0]
```

Lists (II)

Element access read/write

```
>>> v[0]
'hi'
>>> v[0] = 'hey'
>>> v[-1] # last element. Negative = count from the end
>>> v[1:3] # subrange by index (start index, one-beyond)
```

Test if an element is in a list (or not)

```
>>> if 4 in v:
...     print "Found it"
Found it
>>> if 200 not in v:
...     print "Not found"
Not found
```

for and while loops

The `for` statement iterates through a collection, iterable object or generator function.

The `while` statement merely loops until a condition is `False`.

Iterate over list

```
fruits = ["apple", "banana", "cherry"]  
for x in fruits:  
    print(x)
```

Iterate using built-in range function

```
for x in range(0, 3):  
    print "We're on time %d" % (x)
```

Iterate until a condition is met

```
count = 0  
while count < 5:  
    print(count)  
    count += 1    # Same as: count = count + 1
```


Putting lists and loops together is amazing (and complex)

Filter one list into another (the “old” way)

```
newlist = []  
for i in oldlist:  
    if filter(i):  
        newlist.append(function(i))
```

List comprehension (the “pythonic” way)

```
newlist = [function(i) for i in oldlist if filter(i)]
```

where `filter` and `function` just perform “some” operation on the list elements. Basically, the syntax is:

```
[ expression for item in list if conditional ]
```

and this replaces:

```
for item in list:  
    if conditional:  
        expression
```

Useful list comprehension

Filter one list into another (the “old” way)

```
>>> v = [ x*2 for x in range(10) if x % 3 == 0 ]  
>>> v  
[0, 9, 36, 81]
```

List comprehension (the “pythonic” way)

```
newlist = [function(i) for i in oldlist if filter(i)]
```

where filter and function just perform “some” operation on the list elements. Basically, the syntax is:

```
[ expression for item in list if conditional ]
```

and this replaces:

```
for item in list:  
    if conditional:  
        expression
```

Hello Gaussian!

Basic but useful code example

```
import numpy as np
import matplotlib.pyplot as plt

def p(x):
    return np.exp(-x**2)

#let's plot it
x = np.linspace(-3,3,100)
y = p(x)
plt.plot(x,y)
plt.show()
```

Hello Gaussian!

Basic but useful code example

```
import numpy as np
import matplotlib.pyplot as plt
```

```
def p(x):
    return np.exp(-x**2)
```

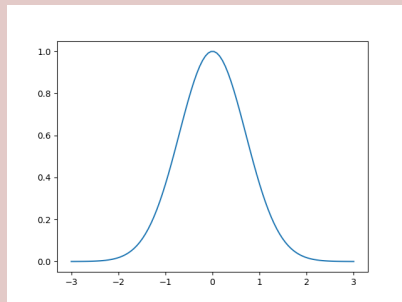
```
#let's plot it
```

```
x = np.linspace(-3, 3, 100)
```

```
y = p(x)
```

```
plt.plot(x, y)
```

```
plt.show()
```



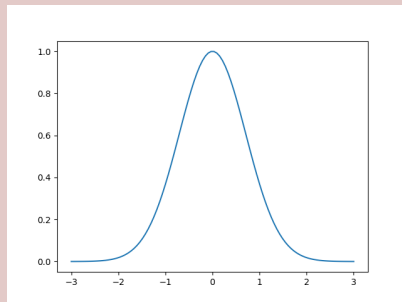
Hello Gaussian!

Basic but useful code example

```
import numpy as np
import matplotlib.pyplot as plt

def p(x):
    return np.exp(-x**2)

#let's plot it
x = np.linspace(-3, 3, 100)
y = p(x)
plt.plot(x, y)
plt.show()
```



But what about that `linspace` thingy? Google it! ([numpy docs](#))

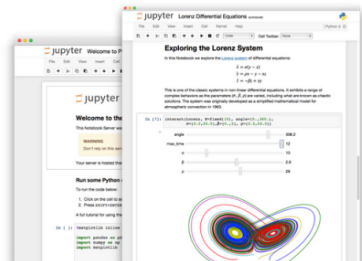
```
numpy.linspace(start, stop, num=50, endpoint=True,
retstep=False, dtype=None)
```

“Returns num evenly spaced samples, calculated over the interval [start, stop].”

Jupyter notebooks

Interactive, web-based, integrated code and documentation environment

We will be following-up with more technical practice with python, but I want to introduce you to the resources that we'll be using this quarter for many of our examples and projects: **Jupyter notebooks**.



The Jupyter Notebook

The Jupyter Notebook is an open-source web application that allows you to create and share documents that contain live code, equations, visualizations and narrative text. Uses include: data cleaning and transformation, numerical simulation, statistical modeling, data visualization, machine learning, and much more.

[Try it in your browser](#)

[Install the Notebook](#)



Language of choice



Share notebooks



Interactive output



Big data integration

Jupyter notebooks on the PHYS 250 computing platform

Built for machine learning applications running mostly in Jupyter.

PHYS 250 Autumn 2018

Home About Services Login

PHYS 250 FALL 2018 COMPUTING PLATFORM

JupyterLab-based infrastructure for Computational Physics

Purpose

A computational platform that supports on-demand JupyterLab instances for interactive python development as well as advanced computational resources such as those required for high-level, compute-intensive machine learning applications.

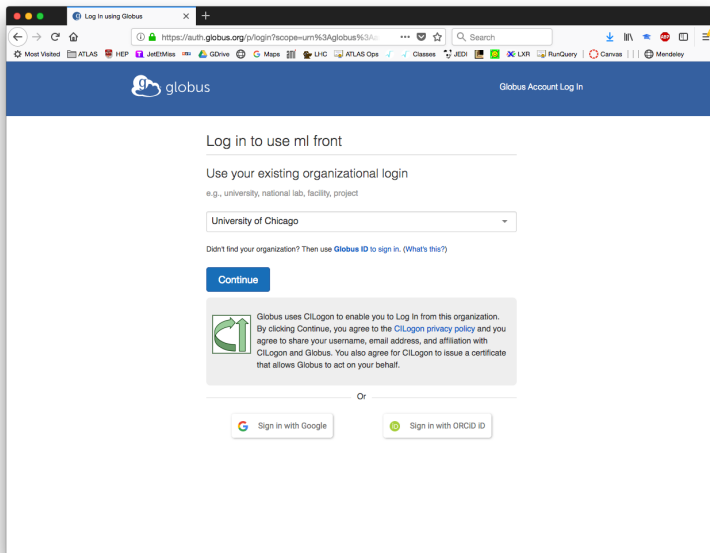
Elements

The platform provides hosted JupyterLab instances for the students in PHYS 250 (Autumn 2018) on GPU resources hosted by the computing center for the ATLAS Experiment

<input checked="" type="checkbox"/> External resources
Intro to Linux (UChicago CSIL)
Intro to Git (UChicago CSIL)
PICUP (Partnership for Integration of Computation into Undergraduate Physics)
Computational Physics text from

Jupyter notebooks on the PHYS 250 computing platform

Built for machine learning applications running mostly in Jupyter.



The screenshot shows a web browser window with the address bar displaying `https://auth.globus.org/p/login?scope=urn%3Aglobus%3A...`. The browser's address bar and tabs are visible at the top. The main content area has a blue header with the Globus logo and a "Log in using Globus" button. Below the header, the text "Log in to use ml front" is displayed. The instructions state: "Use your existing organizational login" and "e.g., university, national lab, facility, project". A dropdown menu shows "University of Chicago" selected. Below this, a link says "Didn't find your organization? Then use Globus ID to sign in. (What's this?)". A blue "Continue" button is present. A box with a green circular arrow icon contains the text: "Globus uses CILogon to enable you to Log In from this organization. By clicking Continue, you agree to the CILogon privacy policy and you agree to share your username, email address, and affiliation with CILogon and Globus. You also agree for CILogon to issue a certificate that allows Globus to act on your behalf." Below this box, the word "Or" is centered. At the bottom, there are two buttons: "Sign in with Google" and "Sign in with ORCID ID".

Jupyter notebooks on the PHYS 250 computing platform

Built for machine learning applications running mostly in Jupyter.

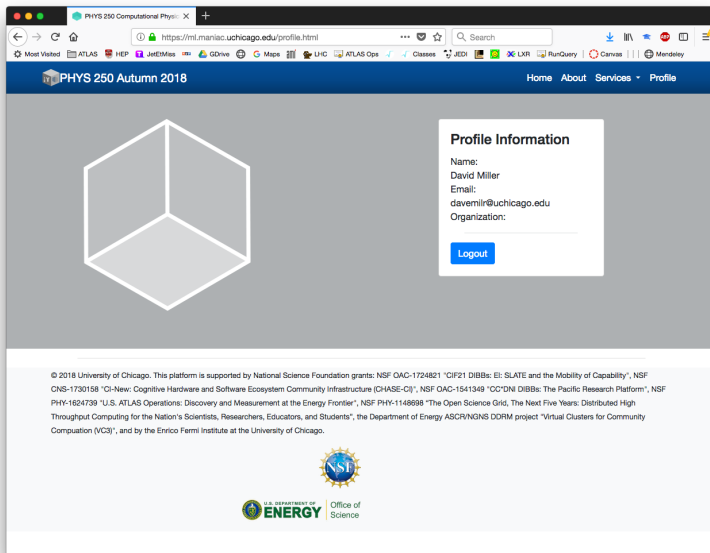
The screenshot shows a web browser displaying the PHYS 250 Autumn 2018 website. The browser's address bar shows the URL <https://ml.maniacs.uchicago.edu/index.html>. The website has a dark blue header with the text "PHYS 250 Autumn 2018" and navigation links: "Home", "About", "Services", and "Profile". The main content area features a large image of a particle accelerator tunnel with the text "PHYS 250 FALL 2018 COMPUTING PLATFORM" and "JupyterLab-based infrastructure for Computational Physics". To the right of this text is a 3D cube icon with a wrench and a screwdriver. Below the main image, there is a "Purpose" section with a paragraph: "A computational platform that supports on-demand JupyterLab instances for interactive python development as well as advanced computational resources such as those required for high-level, compute-intensive machine learning applications." To the right of the "Purpose" section is a table with the following content:

External resources
Intro to Linux (UChicago CSIL)
Intro to Git (UChicago CSIL)
PICUP (Partnership for Integration of Computation into Undergraduate Physics)
Computational Physics text from

Below the "Purpose" section is an "Elements" section with a paragraph: "The platform provides hosted JupyterLab instances for the students in PHYS 250 (Autumn 2018) on GPU resources hosted by the computing center for the ATLAS Experiment".

Jupyter notebooks on the PHYS 250 computing platform

Built for machine learning applications running mostly in Jupyter.



The screenshot shows a web browser window with the URL <https://ml.mariac.uchicago.edu/profile.html>. The browser's address bar and tabs are visible at the top. The website has a blue header with the text "PHYS 250 Autumn 2018" and navigation links: "Home", "About", "Services", and "Profile". The main content area is divided into two sections. On the left is a large, empty white hexagon with a black outline. On the right is a white box titled "Profile Information" containing the following text: "Name: David Miller", "Email: dave.mlr@uchicago.edu", and "Organization:". Below this text is a blue "Logout" button. At the bottom of the page, there is a copyright notice for 2018 University of Chicago, followed by a list of National Science Foundation grants supporting the platform. Below the text are the logos for the National Science Foundation (NSF) and the U.S. Department of Energy Office of Science.

PHYS 250 Autumn 2018

Home About Services Profile

Profile Information

Name:
David Miller

Email:
dave.mlr@uchicago.edu

Organization:

Logout

© 2018 University of Chicago. This platform is supported by National Science Foundation grants: NSF OAC-1724821 "CIF21 DIBBs: EI: SLATE and the Mobility of Capability", NSF CNS-1730158 "CI-New: Cognitive Hardware and Software Ecosystem Community Infrastructure (CHASE-CI)", NSF OAC-1541349 "CC'DNI DIBBs: The Pacific Research Platform", NSF PHY-1624739 "U.S. ATLAS Operations: Discovery and Measurement at the Energy Frontier", NSF PHY-1148698 "The Open Science Grid, The Next Five Years: Distributed High Throughput Computing for the Nation's Scientists, Researchers, Educators, and Students", the Department of Energy ASCR/NGNS DORM project "Virtual Clusters for Community Computation (VC3)", and by the Enrico Fermi Institute at the University of Chicago.

NSF

U.S. DEPARTMENT OF
ENERGY | Office of
Science

Jupyter notebooks on the PHYS 250 computing platform

Built for machine learning applications running mostly in Jupyter.

The screenshot shows a web browser window with the URL <https://ml.maniac.uchicago.edu/profile.html>. The page has a blue header with the text "PHYS 250 Autumn 2018" and navigation links: "Home", "About", "Services", and "Profile". A large white hexagonal logo is on the left. On the right, a "Profile Information" box displays the following details:

- Name: David Miller
- Email: davemilr@uchicago.edu
- Organization:

Below the profile information is a blue "Logout" button. To the right of the profile box is a dropdown menu with the following options:

- My Services
- Monitor
- Public Lab
- Users

At the bottom of the page, there is a copyright notice for 2018 University of Chicago, supported by various National Science Foundation grants. Below the text are the logos for the National Science Foundation (NSF) and the U.S. Department of Energy Office of Science.

Jupyter notebooks on the PHYS 250 computing platform

Built for machine learning applications running mostly in Jupyter.

PHYS 250 Computational Physics X

https://ml.maniac.uchicago.edu/services.html

PHYS 250 Autumn 2018

Home About Services Profile

Your deployments

From this page you can manage your private services. Please delete a service when you don't need it anymore. To check if a service status changed simply refresh this page.

[New Personal JupyterLab](#)

Running Services

Service	Name	Started at	Ending at	GPUs	Cores	Memory	Link	Status	Command
No data available in table									

Showing 0 to 0 of 0 entries

All Services

Service	Name	Started at	Ending at	GPUs	Cores	Memory
No data available in table						

Showing 0 to 0 of 0 entries

© 2018 University of Chicago. This platform is supported by National Science Foundation grants: NSF OAC-1724821 "CIF21 DIBBs: EI: SLATE and the Mobility of Capability", NSF CNS-1730158 "CI-New: Cognitive Hardware and Software Ecosystem Community Infrastructure (CHASE-CI)", NSF OAC-1541349 "CC'DNI DIBBs: The Pacific Research Platform", NSF PHY-1624739 "U.S. ATLAS Operations: Discovery and Measurement at the Energy Frontier", NSF PHY-1148698 "The Open Science Grid, The Next Five Years: Distributed High Throughput Computing for the Nation's Scientists, Researchers, Educators, and Students", the Department of Energy ASCR/NGNS DDRM project "Virtual Clusters for Community Collaboration (VCC)" and by the Evron Erami Institute at the University of Chicago.

Jupyter notebooks on the PHYS 250 computing platform

Built for machine learning applications running mostly in Jupyter.

The screenshot shows a web browser window with the URL `https://ml.maniac.uchicago.edu/PrivateJupyter.html`. The page has a dark blue header with the text "PHYS 250 Autumn 2018" and navigation links: "Home", "About", "Services", and "Profile".

The main content area is divided into two columns. The left column is titled "Private JupyterLab" and contains the text: "Use this for private code development or if you need dedicated resources." Below this is a section titled "Instructions (read first)" with a bulleted list:

- Fill out the form to the right.
- Upon submission a dedicated JupyterLab instance will be spawned in the background
- You'll receive a JupyterLab link to be used once the notebook has been scheduled (this can take several minutes, or longer if the resources are busy)
- We suggest organizing your notebook in GitHub and cloning it manually once your notebook starts up. Remember to push any changes before the notebook expires.

The right column is titled "Configure your JupyterLab instance" and contains the text: "Please only select what you actually need." Below this are several form fields:

- Name ***: A text input field containing "Lecture 1".
- Password ***: A password input field with four asterisks "****".
- Expiration of your JupyterLab [days] ***: A dropdown menu with "1" selected.
- GPUs**: A dropdown menu with "0" selected.
- CPUs**: A dropdown menu with "1" selected.
- Memory [GB]**: A dropdown menu with "8" selected.
- Check out a GitHub repo (use full URL including ".git")**: An empty text input field.

At the bottom of the right column is a blue button labeled "START".

Jupyter notebooks on the PHYS 250 computing platform

Built for machine learning applications running mostly in Jupyter.

The screenshot shows a web browser window with the URL <https://ml.maniac.uchicago.edu/services.html>. The page title is "PHYS 250 Autumn 2018". The navigation bar includes links for Home, About, Services, and Profile. The main content area is titled "Your deployments" and contains a blue button labeled "New Personal JupyterLab". Below this is a section titled "Running Services" which displays a table of active JupyterLab instances. The table has columns for Service, Name, Started at, Ending at, GPUs, Cores, Memory, Link, Status, and Command. One instance is listed: "Private JupyterLab" with name "lecture-1", started on Tue, 02 Oct 2018 at 14:31:14 GMT, and ending on Wed, 03 Oct 2018 at 14:31:14 GMT. Below the table, it says "Showing 1 to 1 of 1 entries". There is also a section titled "All Services" which displays a table of all JupyterLab instances. This table has columns for Service, Name, Started at, Ending at, GPUs, Cores, and Memory. It lists five instances: "lecture-1", "instructor-lab", "test-3", "my-lab 2", and "my-lab 2".

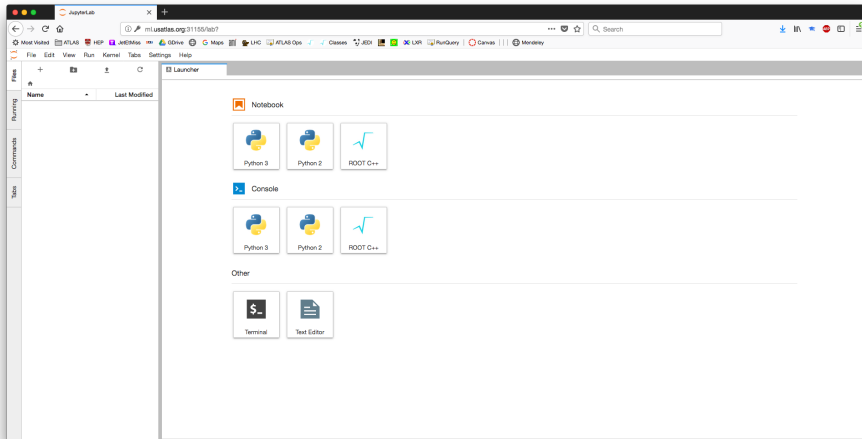
Service	Name	Started at	Ending at	GPUs	Cores	Memory	Link	Status	Command
Private JupyterLab	lecture-1	Tue, 02 Oct 2018 14:31:14 GMT	Wed, 03 Oct 2018 14:31:14 GMT	0	1	8Gi	http://ml.usatlas.org:31155	Running	Delete

Showing 1 to 1 of 1 entries

Service	Name	Started at	Ending at	GPUs	Cores	Memory
Private JupyterLab	lecture-1	Tue, 02 Oct 2018 14:31:14 GMT	Wed, 03 Oct 2018 14:31:14 GMT	0	1	8
Private JupyterLab	instructor-lab	Mon, 01 Oct 2018 18:40:41 GMT	Tue, 02 Oct 2018 18:40:41 GMT	0	1	8
Private JupyterLab	test-3	Fri, 28 Sep 2018 20:07:37 GMT	Sat, 29 Sep 2018 20:07:37 GMT	0	1	8
Private JupyterLab	my-lab 2	Fri, 28 Sep 2018 19:59:02 GMT	Sat, 29 Sep 2018 19:59:02 GMT	0	1	8
Private JupyterLab	my-lab 2	Fri, 28 Sep 2018 19:58:19 GMT	Sat, 29 Sep 2018 19:58:19 GMT	0	1	8

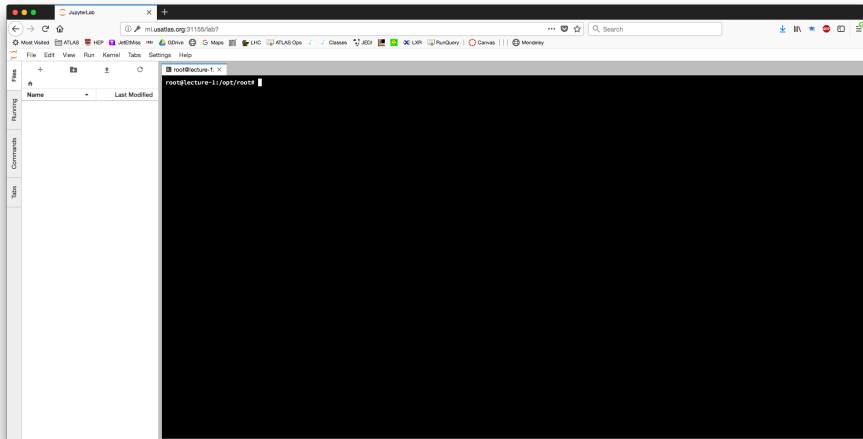
Jupyter notebooks on the PHYS 250 computing platform

Built for machine learning applications running mostly in Jupyter.



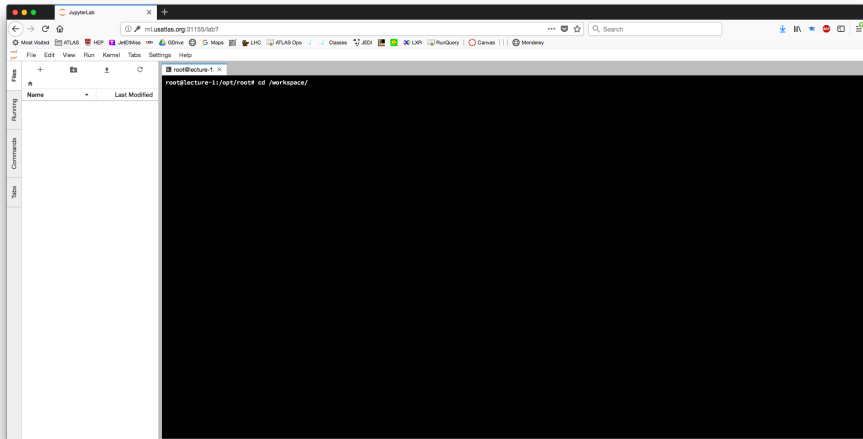
Jupyter notebooks on the PHYS 250 computing platform

Built for machine learning applications running mostly in Jupyter.



Jupyter notebooks on the PHYS 250 computing platform

Built for machine learning applications running mostly in Jupyter.



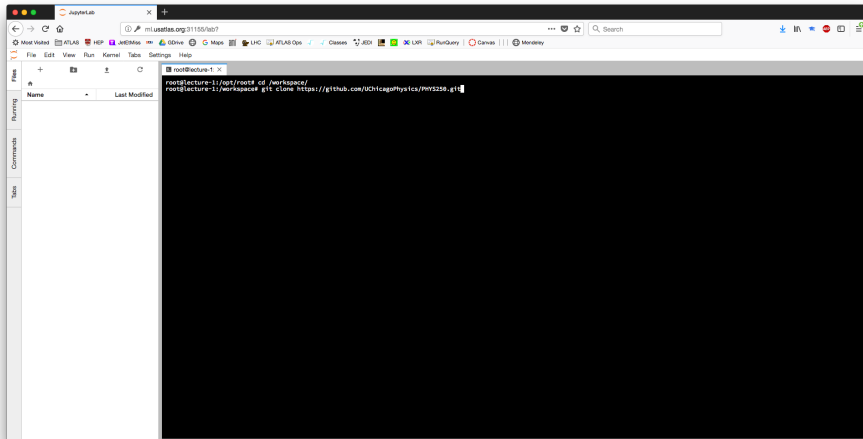
Jupyter notebooks on the PHYS 250 computing platform

Built for machine learning applications running mostly in Jupyter.

The screenshot displays the GitHub interface for the repository `UChicagoPhysics/PHYS250`. The repository is described as the "University of Chicago PHYS 250 Computational Physics software repository". It shows 20 commits, 1 branch, 0 releases, and 1 contributor. A sidebar on the right provides options to clone the repository using HTTPS or SSH, or to download the ZIP file. The main content area lists files and their last update times: `Examples` (4 days ago), `LearningGoals` (4 days ago), `Slides` (19 hours ago), `Syllabus` (18 hours ago), `gIgitgnore` (4 days ago), `README.md` (19 hours ago), and `global.sty` (18 hours ago). The `README.md` file is selected, showing the title "PHYS 250: Computational Physics" and a welcome message.

Jupyter notebooks on the PHYS 250 computing platform

Built for machine learning applications running mostly in Jupyter.



Jupyter notebooks on the PHYS 250 computing platform

Built for machine learning applications running mostly in Jupyter.

